



AKÇADAĞ MYO

C PROGRAMLAMA

1. HAFTA



BU HAFTA

- I. C Dili
- II. Değişken ve Sabit Kavramı
- III. Operatörler
- IV. Başlık Dosyaları

C DİLİ

C programlama dili, Bell Laboratuvarları'nda çalışmakta olan Dennis Ritchie ve Ken Thompson tarafından, Unix sistemlerine programlama yapmak amacıyla 1969-1973 yılları arasında geliştirilir.

C'nin bu derece ilgi görmesinde, dil hakkında Dennis Ritchie ve Brian Kernighan'ın 1978 yılında yazdığı "C Programlama Dili" isimli kitabın etkisi büyüktür. Bu kitap; C dili, ANSI (Amerikan Ulusal Standartlar Enstitüsü) tarafından standartlaştırılana dek tek referans kaynağı olarak kullanılır.



Ken Thompson (solda) ile Dennis Ritchie (sağda)



C DİLİ

C Makine Dili ile üst düzey programlama dili arasında olan son derece esnek kullanımlı bir programlama dilidir. Esnek olması yazacağınız programda daha dikkatli olmanız gerektiği anlamına gelir. Unutulacak bir işaret belki derleyici hatasına neden olmayacaktır ancak programınızı da doğru çalıştırmayacaktır.

C++



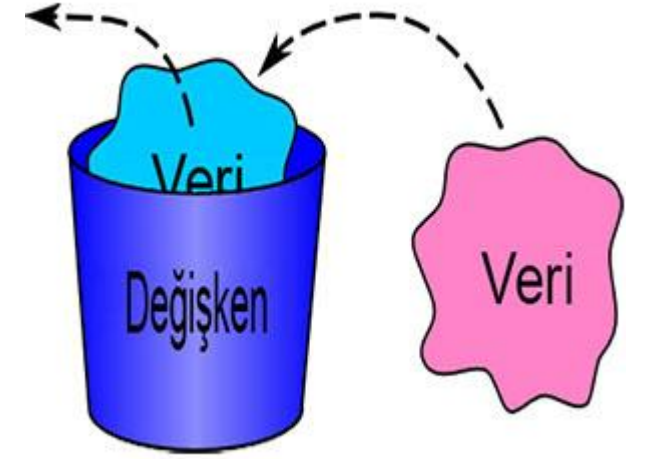
DEV C++

C komutlarını yazarak, derleyip çalıştırabileceğimiz birçok derleyici program vardır. Biz dersimizde Bloodshed yazılım firmasının geliştirdiği DevC++ 5.11 sürümünü kullanacağız. Söz konusu sürümün, şu ana dek hazırlanmış en güncel C++ sürümü olmasının yanı sıra; ücretsiz olması, Türkçe menülere sahip bulunması, kurulum gerektirmediği için USB flash belleklerden dahi kolayca çalıştırılabilmesi, hem 32 bit hem de 64 bit işletim sistemleri üzerinde sorunsuz olarak görev yapan dahili derleyicisiyle birlikte kullanılabilmesi ve ileri programlama tekniklerinin programa dahil edilebilmesi için devpaks.org sitesinden ücretsiz WebUpdate hizmeti alabilmesi gibi önemli avantajları vardır.



DEĞİŞKENLER

Değişkenler istenilen değerlerin belirli bir süre hafızada tutulması için kullanılırlar. Tanımlanan her değişken bellekte bir miktar alan kaplar. Değişkene atanan değer programın farklı aşamalarında farklı değerler alabilir, içeriği sabit değildir. Doğal olarak değişkene yeni bir değer atandığında eskisi kaybolacaktır. Bir değişken programlar için çok önemlidir. Çünkü matematiksel, alfabetik vb. tüm işlemler bu değişkenler üzerinde yapılır ve sonuçlar bu değişkenler üzerinde tutulur.





DEĞİŞKEN ADLARI

Değişken adı verirken uyulması gereken bazı kurallar vardır. Bu kurallara uyulsa dahi değişken isimleri verirken yazılımcı mantıklı ve değişkenin adına baktığında hangi değeri tuttuğunu hatırlayacağı isimleri seçmesi mantıklı olacaktır. Örneğin yazılı, sözlü ve ortalama değerlerini tutacak değişkenler için a,b,c yerine yazili, sozlu, ortalama gibi değişken adlarının kullanılması daha sağlıklı olacaktır.



C DİLİNDE DEĞİŞKEN ADI KURALLARI

1. Değişken isimlerinde Türkçe karakterler (ç, ğ, ü, ö, ş, ı, İ) kullanılamaz.

Doğru: sinavnotu

Yanlış: sinavnotu

2. Değişkenler içerisinde altçizgi _ dışında herhangi bir özel karakter kullanılamaz.

Doğru: adi_soyadi

Yanlış: adi+soyadi

3. Değişkenler içerisinde rakam bulunabilir fakat rakam değişken isminin en başında olamaz.

Doğru: yazili1

Yanlış: 1.yazili

4. Değişkenler içerisinde boşluk karakteri kullanılamaz, çoğunlukla boşluk yerine _ kullanılır.

Doğru: vize_notu

Yanlış: vize notu



C DİLİNDE DEĞİŞKEN ADI KURALLARI

5. Değişken isimlerinde büyük küçük harf kuralı vardır. **Vize** ile **vize** değişkenleri farklı değişkenlerdir ve program içerisinde her biri farklı değerler alabilir.
6. C programında kullanılan özel komutlar (auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while) değişken adı olamazlar.
7. Bir değişken ismi en fazla 255 karakterden oluşur. 255 karakterden daha uzun bir değişken ismi olamaz.



C DİLİNDE DEĞİŞKEN TİPLERİ

C dilinde kullanılan değişken tiplerini iki sınıfa ayırabiliriz:

➤ Alfa sayısal

➤ Sayısal



ALFA SAYISAL DEĞİŞKENLER

Sayısal olmayan değişkenleri tutmak için kullanılan değişkenlerdir.

char degisken_adi;

şeklinde tanımlanırlar. C dilinde bu şekilde tanımladığınız bir alfa sayısal değişken içerisinde sadece tek bir harf saklayabilirsiniz.

Eğer birden fazla harf saklanması gerekiyorsa şu tanımlı kullanmanız daha uygun olacaktır:

char degisken_adi[28];

Bu tanımda ise toplam 28 karakter uzunluğunda adı isim olan bir alfa sayısal değişken tanımlanmıştır. Her bir karakter bellekte 1 baytlık alan kaplar.



SAYISAL DEĞİŞKENLER

Sayısal ifadeleri tutmak için kullanılan değişkenlerdir. Tam sayılar için int, ondalıklı sayılar için float ve double, mantıksal ifadeler için bool değişkeni kullanılır. Int, float ve double değişkenleri long, short, unsigned gibi parametrelerle kullanılabilir. Int tipi değişkenler 2-4 bayt, float ve double tipi değişkenler 4-8 bayt, bool tipi değişken ise 1 baytlık alan kaplar.

x = 3





TAMSAYI (INT)

Değişken Tipi	Tanım	Sınırlar
int	Tamsayı	-32.768 ile +32.767
shortint	Kısa Tamsayı	-32.768 ile +32.767
longint	Uzun Tamsayı	-2.147.483.648 ile +2.147.483.647
unsigned int	İşaretsiz Tamsayı	0 ile +4.294.967.295
unsigned shortint	İşaretsiz Kısa Tamsayı	0 ile 65.535
unsigned longint	İşaretsiz Uzun Tamsayı	0 ile +4.294.967.295



TAMSAYI (INT)

Tam sayı değeri alabilecek değişkenleri tanımlamak için int değişkeni kullanılır. Programın başında tanımlanabildiği gibi dışarıdan değer istenerek te değişkene değer ataması yapılır. Eğer bir değişkenin değeri başka 2 değere göre atanacaksa ilk değer mutlaka atanmalıdır. Aksi halde derleyici değişkene kendisi bir değer verecek ve yazılan program yanlış çalışacaktır.

```
int degisken_adi;
```

```
degisken_adi=5;
```

```
int sayi1=6;
```

```
int sayi1,sayi2,sayi3;
```



ONDALIKLI SAYI (FLOAT, DOUBLE)

Değişken Tipi	Tanım	Sınırlar
float	Kayan Ondalıklı Sayı	$3,4 \times 10^{-38}$ ile $3,4 \times 10^{38}$
double	Çift Ondalıklı Sayı	$1,7 \times 10^{-308}$ ile $1,7 \times 10^{308}$
longdouble	Uzun Ondalıklı Sayı	$1,2 \times 10^{-4932}$ ile $1,2 \times 10^{4932}$



ONDALIKLI SAYI (FLOAT, DOUBLE)

Ondalıkli sayı değeri alabilecek değişkenleri tanımlamak için float veya double değişkeni kullanılır. Programın başında tanımlanabildiği gibi dışarıdan değeri istenerek te değişkene değeri ataması yapılır. Sayılar arasına , (virgül) değil . (nokta) işareti konulur.

float degisken_adi;

degisken_adi=5.365;

double sayi1=6.64;

float sayi1,sayi2,sayi3;



SABİTLER

Değeri program boyunca değişmeyecek ve aynı kalacak durumlar için sabitler kullanılır. Sabitler bellekte değişkenlerden çok daha az yer kapladıkları için tercih edilirler. Sabitler program başında tanımlanır ve bu tanımlanan değerden başka bir değer alamazlar. C dilinde sabitler `#define` komutu ile tanımlanır.

`#define` `sabit_adi` `alacagi_değer`;

`#define` `PI` `3.14`;

`#define` `harf` `"a"`;

Sabitler tanımlanırken tamsayı, ondalıklı sayı veya alfabetik karakterler kullanılabilir. Sabit tanımlamalarında değişken tiplerini belirtmeye gerek yoktur.



OPERATÖRLER

Kullanılan değişkenler, değerler veya sabitler üzerinde matematiksel ve mantıksal işlemleri gerçekleştirmek için kullanılan özel karakterlere operatör adı verilir. C dilinde kullanılan operatörler 5 grupta incelenebilir:

- Aritmetiksel Operatörler
- Atama Operatörleri
- Karşılaştırma Operatörleri
- Mantıksal Operatörler
- Unary Operatörler



ARİTMETİKSEL OPERATÖRLER

Matematiksel işlemlerin yapıldığı operatörlerdir. 4 işlem operatörü tıpkı diğer programlama dillerinde olduğu gibi kullanılır. Ancak int tipindeki 2 sayı / operatörü ile bölündüğünde sonuç tamsayı olarak bulunur. Eğer bölme işlemi sonucu kesirli bir sayı olacaksa sonuç float tipine dönüştürülmelidir.

Operatör	İşlem	Kullanımı
-	Çıkarma	$a-b$
+	Toplama	$a+b$
*	Çarpma	$a*b$
/	Bölme	a/b
%	Mod alma	$a\%b$



ATAMA OPERATÖRLERİ

C dilinde değişkenlere değer atama işlemlerinde = operatörü kullanılır. Eğer bir matematiksel işlemin sonucuna göre değer ataması yapılacaksa = işaretinin başına aritmetiksel operatörler eklenerek atama işlemi gerçekleştirilir.

Operatör	İşlem	Kullanımı	Anlamı
=	Atama	$x=7$	$x=7$
+=	Ekleyerek Atama	$x+=3$	$x=x+3$
-=	Eksilterek Atama	$x-=3$	$x=x-3$
=	Çarparak Atama	$x=3$	$x=x*3$
/=	Bölerek Atama	$x/=3$	$x=x/3$
%=	Mod Alarak Atama	$x\%=3$	$x=x\%3$



KARŞILAŞTIRMA OPERATÖRLERİ

Genellikle koşullu ifadelerle kullanılan iki değeri, değişkeni veya sabiti diğer değer, değişken veya sabitlerle karşılaştırmak için kullanılır. Karşılaştırma operatörlerinde kurulan şart cümlesi doğru ise 1 (true), yanlış ise 0 (false) değeri üretilir.

Operatör	İşlem	Kullanımı
==	Eşit	$x==y$
!=	Eşit Değil	$x!=y$
<	Küçük	$x<y$
<=	Küçük veya Eşit	$x<=y$
>	Büyük	$x>y$
>=	Büyük veya Eşit	$x>=y$



MANTIKSAL OPERATÖRLER

Bu operatörler birden fazla karşılaştırmanın veya karşılaştırmanın olumsuzunun yapılacağı durumlarda kullanılır. AND mantıksal operatörüyle birleştirilen koşulların tamamının sağlandığı durumlarda true değeri, koşullardan birisi dahi sağlanmazsa false değeri üretir. OR mantıksal operatöründe koşullardan bir tanesi bile sağlanırsa true, hiçbir koşul sağlanmazsa false değeri üretir. NOT mantıksal operatöründe ise koşul doğruysa true, yanlışsa false değeri üretir.

Operatör	İşlem	Kullanımı
!	NOT	!(x==y)
&&	AND	x=3 && y>5
and	AND	x=3 and y>5
	OR	x=3 y>5
or	OR	x=3 or y>5



UNARY OPERATÖRLERİ

Unary operatörleri diğer programları dillerinde olmayan ve C diline özgü operatörlerdir. ++ ve – olmak üzere iki unary operatörü vardır. Bu operatörlerin değişkenin başında veya sonunda kullanılması arasında farklılık vardır.

Operatör	İşlem	Kullanımı
++	1 arttırma	x++ veya ++x
--	1 eksiltme	x-- veya --x



UNARY OPERATÖRLERİ

`x++` ile `++x` kullanımı arasında bir farklılık vardır. Örneğin;

`x = 10; y = x++;`

Satırları çalıştırıldığında;

`x = 11; y = 10` değerleri elde edilir. Yukarıdaki ifade de `y` değişkenine önce `x` değişkenindeki değer aktarılır daha sonra `x` değişkeni 1 arttırılır.

`x = 10; y = ++x;`

Satırları çalıştırıldığında;

`x = 11; y = 11` değerleri elde edilir. Yukarıdaki ifade de önce `x` değişkeni 1 arttırılır sonra `y` değişkenine `x` değişkenindeki değer aktarılır.



UNARY OPERATÖRLERİ

Aşağıda değerleri verilen değişkenlerin son değerlerini hesaplayınız.

1. $x = 5$ $y = ++x$ $z = --y$ $t = z++$



UNARY OPERATÖRLERİ

Aşağıda değerleri verilen değişkenlerin son değerlerini hesaplayınız.

2. $x = 6$ $y = x--$ $z = y++$ $t = ++x$



UNARY OPERATÖRLERİ

Aşağıda değerleri verilen değişkenlerin son değerlerini hesaplayınız.

3. $x = 3$ $y = --x$ $z = --x$ $t = ++x$



UNARY OPERATÖRLERİ

Aşağıda değerleri verilen değişkenlerin son değerlerini hesaplayınız.

4. $x = 1$ $y = x++$ $z = --y$ $t = z++$



UNARY OPERATÖRLERİ

Aşağıda değerleri verilen değişkenlerin son değerlerini hesaplayınız.

5. $x = 2$ $y = ++x$ $z = y--$ $t = --z$



İŞLEMLERİN ÖNCELİK SIRASI

C dilinde işlemlerin düzenli bir şekilde yürütebilmesi amacıyla her bir işlem için bir öncelik sırası tanımlanmıştır. Aynı önceliğe sahip işlemlerde öncelik sırası soldan sağa doğrudur.

ÖNCELİK SIRASI	OPERATÖR	AÇIKLAMA
1	()	Parantez
2	x++	Sonra arttır
	x--	Sonra elstir
3	++x	Önce arttır
	--x	Önce elstir
	!	NOT
4	*	Çarpma
	/	Bölme
	%	Mod alma
5	+	Toplama
	-	Çıkarma
6	>	Büyük
	<	Küçük
	>=	Büyük veya eşit
	<=	Küçük veya eşit
7	==	Eşit
	!=	Eşit değil
8	&&	AND
		OR
9	=	Atama



BAŞLIK DOSYALARI

C programında kullanılan her komut bir başlık dosyasında bulunur ve program yazımı esnasında kullanılan her komutun ait olduğu başlık dosyası programın başında `#include` deyimiyle açılır. Örneğin, C'de giriş/çıkış fonksiyonları için gerekli olan komutlar `stdio.h` başlık dosyası içindedir ve şu şekilde programa dahil edilir:

`#include <stdio.h>`

Eğer kullandığınız komutlara ait başlık dosyaları açılmazsa program derleme esnasında hata mesajı verecektir.



BAŞLIK DOSYALARI

C dilinde çok sık kullanılan başlık dosyaları şunlardır:

- ctype.h
- stdio.h
- conio.h
- stdlib.h
- string.h
- math.h
- time.h
- graphics.h



cctype.h

Bu başlık dosyasında karakterlerle ilgili komutları çalıştıran rutinler yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
isalpha	Bir karakterin alfabetik olup olmadığını kontrol eder.
isalnum	Bir karakterin nümerik veya alfabetik olup olmadığını kontrol eder.
isascii	Bir karakterin ASCII karakter olup olmadığını kontrol eder.
isdigit	Bir karakterin nümerik olup olmadığını kontrol eder.
islower	Bir karakterin küçük harf olup olmadığını kontrol eder.
isupper	Bir karakterin büyük harf olup olmadığını kontrol eder.
toascii	ASCII kodu girilen karakteri görüntüler.
tolower	Girilen karakteri küçük harfe çevirir.
toupper	Girilen karakteri büyük harfe çevirir.



stdio.h

Bu başlık dosyasında temel giriş/çıkış komutlarını yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
printf	Değişkenlerin tuttuğu değerleri, onların adreslerini veya bir mesajı ekranda yazdırmak için kullanılır.
getchar	Standart girişten bir karakter okunur. Programı istenen bir yerde durdurup, bir karakter girinceye kadar bekletir.
puts	Program sonucunu ekranda görebilmemizi sağlar. Ekran bu karakter topluluğu yazdıktan sonra, imleci alt satıra geçirir.
scanf	Klavyeden veri okumak için kullanılır.
gets	Klavyeden bir karakter topluluğunu okumak için kullanılır.



conio.h

Bu başlık dosyasında klavye ve ekran kullanımını kolaylaştıran komutlar yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
getch	Klavyeden yazılan tek bir karakteri okur.
getche	karakter değerini girer girmez enter tuşunun basılmasını beklemeden okunan değeri döndürür.



stdlib.h

Bir programı sonlandıran, dinamik bellek yönetiminde kullanılan veya rastgele sayı üreten komutlar bu başlık dosyası altında bulunur. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
exit	Programı sonlandırarak kontrolü işletim sistemine bırakır.
atoi	Bir karakter topluluğunu tamsayıya çevirir.
atol	Bir karakter topluluğunu uzun tamsayıya çevirir.
atof	Bir karakter topluluğunu ondalıklı sayıya çevirir.
rand	Rastgele sayı değeri üretir.



string.h

Metin dizileriyle işlem yapmamızı sağlayan komutlar bu başlık dosyasında yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
strcat	Bir metin dizisini diğer bir metin dizisinin sonuna ekler.
strcmp	Verilen iki metin dizisini karşılaştırır.
strcpy	Bir metin dizisini kopyalar.
strlen	Bir metin dizisinin uzunluğunu hesaplar.
strlwr	Bir metin dizisinin karakterlerini küçük harfe çevirir.
strupr	Bir metin dizisinin karakterlerini büyük harfe çevirir.



math.h

Bu başlık dosyasında matematiksel işlemlerle ilgili komutlar yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
abs	Bir sayının mutlak değerini hesaplar.
floor	Kesirli bir sayıyı aşağı yuvarlar.
ceil	Kesirli bir sayıyı yukarı yuvarlar.
sqrt	Bir sayının karekökünü hesaplar.
pow	Bir sayının üssünü almak için kullanılır.
log10	Bir sayının 10 tabanında logaritmasını hesaplar.
sin	Bir sayının sinüsünü hesaplar.
cos	Bir sayının kosinüsünü hesaplar.
tan	Bir sayının tanjantını hesaplar.



time.h

Bu başlık dosyasında zaman ve tarihle ilgili komutlar yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
asctime	Gün, ay, ayın günü, saat, dakika, saniye, yıl formatında tarih bilgisini gösterir.
ctime	Gün, ay, ayın günü, saat, dakika, saniye, yıl formatında tarih bilgisini gösterir.
difftime	İki zaman arasındaki farkı saniye cinsinden hesaplar.
gmtime	Greenwich'e göre tarih bilgisini gösterir.
localtime	Yerel saat dilimine göre tarih bilgisini gösterir.
mktime	Tarih ve saat bilgisini gösterir.



graphics.h

Bu başlık dosyasında grafiklerle ilgili komutlar yer almaktadır. Aşağıdaki tabloda bu başlık dosyasında sık kullanılan komutlar ve açıklamaları verilmiştir.

Komut	Açıklama
line	Düz bir çizgi çizer.
circle	Daire çizer.
arc	Yay çizer.
rectangle	Dikdörtgen çizer.
setcolor	Çizilen bir şeklin rengini belirler.
move	Noktayı taşır.